Continuous restricted Boltzmann machine with an implementable training algorithm

H. Chen and A.F. Murray

Abstract: The authors introduce a continuous stochastic generative model that can model continuous data, with a simple and reliable training algorithm. The architecture is a continuous restricted Boltzmann machine, with one step of Gibbs sampling, to minimise contrastive divergence, replacing a time-consuming relaxation search. With a small approximation, the training algorithm requires only addition and multiplication and is thus computationally inexpensive in both software and hardware. The capabilities of the model are demonstrated and explored with both artificial and real data.

1 Introduction

Probabilistic generative models offer flexible data modelling, wherein stochasticity both represents the natural variability of real data and drives a search of solution space during training. Our primary interest is in processing and modelling continuous data close to a sensor interface. It is therefore important that such models are amenable to, or at least not resistant to, analogue or mixed-mode VLSI implementation. The product of experts (PoE) [1] has been shown to be a flexible architecture and 'minimising contrastive divergence' (MCD) can underpin a simple training rule [2]. The restricted Boltzmann machine (RBM) [3] with an MCD rule has been shown to be amenable to further simplification and use with real, continuous data [4]. However, the binary RBM is not ideally suited to modelling continuous data, even with pragmatic modifications made to the data representation [2, 4]. The probabilistic units are binary-stochastic and approximations must be made to encode continuous data. In the rate-coded RBM (RBMrate) [5], stochastic units can adopt intermediate discrete values, and thus have greater modelling flexibility. However, repeated, uncorrelated sampling is awkward in both hardware and software. The diffusion network (DN) [6] with symmetric connections can be shown to be a continuous Boltzmann machine. We introduce a form of continuous restricted Boltzmann machine (CRBM) [7], by applying the MCD rule to a constrained diffusion network. We also develop a simplification of the rule that renders it more amenable to implementation. With continuous-valued stochastic units, the CRBM offers improved modelling ability with both artificial and real (in our example, ECG) continuous data.

IEE Proceedings online no. 20030362

DOI: 10.1049/ip-vis:20030362

Paper first received 10th April and in revised form 5th December 2002

The authors are with the School of Engineering and Electronics, University of Edinburgh, Mayfield Rd., Edinburgh EH9 3JL, UK

2 Binary and discrete restricted Boltzmann machine

The RBM has one hidden and one visible layer with only interlayer connections [3]. Fig. 1 shows an RBM with two visible units, four hidden units and two (permanently-on) bias units v_0 and h_0. The visible and hidden units have binary states $\{0, 1\}$ and are connected by weight matrix $\{w\}$. Let v_i and h_j represent the states of visible unit i and hidden unit j, respectively, and $w_{ij} = w_{ji}$ the bidirectional weights. The state probabilities of the units are

$$p_{vi} = p(v_i = 1) = \frac{1}{1 + \exp(-\sum_j w_{ij} h_j)} \text{ and}$$
$$p_{hj} = p(h_j = 1) = \frac{1}{1 + \exp(-\sum_i w_{ij} v_i)} \quad (1)$$

As no intralayer connections exist, units within a layer are conditionally independent and can also be updated in parallel. The MCD training rule for an RBM replaces the computationally expensive relaxation search of the Boltzmann machine [8] with a single step of Gibbs sampling [2]. In MCD training, a training datum is first presented to the visible units to produce $\{v_i\}$. The binary hidden states $\{h_j\}$ are then sampled according to probabilities in (1). Repeating this process once more to update the visible and then the hidden units produces

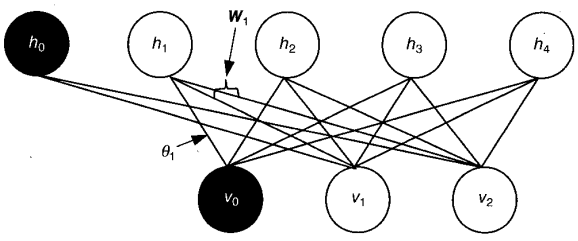

Fig. 1 *Diagram of an RBM with two visible units and four hidden units*

one-step 'reconstructed' states $\{\hat{v}_i\}$ and $\{\hat{h}_j\}$. The MCD update equation for w_{ij} is

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle - \langle \hat{v}_i \hat{h}_j \rangle) \tag{2}$$

where $\langle \cdot \rangle$ refers to the mean over the training data.

Following Hinton in [2], the RBM is able to model grey-level handwritten digits [2] and continuous data [4], if the probabilities in (1) of visible units are treated as approximations to the continuous values, and the probabilities p_{vi} and p_{hj} are substituted for v_i and h_j in (2). However, the RBM with binary hidden units tends to generate continuous data with high symmetry. For example, if an RBM with two hidden units is trained to generate data in a training set, only hidden states $(h_1, h_2) = (0, 0), (0, 1), (1, 0)$ and $(1, 1)$ can occur. As a consequence, even if training data only exist in the regions modelled by (0, 0), (0, 1) and (1, 0), data will be generated in the area modelled by (1, 1), albeit with lower probability. The limitation above has been made clear experimentally in [7].

The 'rate-coded RBM' (RBMrate) [5] alleviates this limitation by estimating the 'firing rate' of a binary-stochastic unit. This estimate is achieved by sampling the unit's state m times, representing continuous values by the number of active samples. The states $\{v_i\}$ $\{h_j\}$ of RBMrate units thus have discrete values of $\{1, 2, 3, \ldots, m\}$, but. the simple training algorithm of (2) is preserved without loss of rigour. RBMrate offers an improved ability to model continuous image data [5], but the repetitive sampling of RBMrate is inconvenient and will exacerbate noise in the power supplies of a VLSI implementation, placing the circuits in danger of synchronisation [9]. We have therefore chosen Movellan's method in a diffusion network [6] to introduce equivalent stochastic behaviour.

3 Continuous restricted Boltzmann machine (CRBM)

3.1 Continuous stochastic unit

The repeated sampling procedure in RBMrate creates input noise, giving rise to a discrete-valued output with a binomial distribution. Following this lead, we introduce a continuous stochastic unit by adding a zero-mean Gaussian noise to the input of a sampled sigmoid unit. Let s_j be the output of neuron j, with inputs from neurons with states $\{s_i\}$. (From Section 3, s_j represents v_j for a visible neuron, and h_j for a hidden neuron.)

$$s_j = \varphi_j\left(\sum_i w_{ij} s_i + \sigma \cdot N_j(0, 1)\right) \tag{3}$$

with

$$\varphi_j(x_j) = \theta_L + (\theta_H - \theta_L) \cdot \frac{1}{1 + \exp(-a_j x_j)} \tag{4}$$

where $N_j(0, 1)$ represents a Gaussian random variable with zero mean and unit variance. The constant σ and $N_j(0, 1)$ thus constitute a noise input component $n_j = \sigma \cdot N_j(0, 1)$ according to a probability distribution

$$p(n_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-n_j^2}{2\sigma^2}\right) \tag{5}$$

$\varphi_j(x)$ is a sigmoid function with asymptotes at θ_L and θ_H. Parameter a_j controls the slope of the sigmoid function, and thus the nature and extent of the unit's stochastic behaviour [7]. Such behaviour is similar to the noisy unit in [10], where the variance of the added noise is tuned. Replacing the binary stochastic unit in RBM by this continuous form of stochastic unit leads to a continuous RBM (CRBM), within which a_j is a 'noise-control' parameter, allowing a smooth transition from noise-free, deterministic behaviour (small a_j) to binary-stochastic behaviour (large a_j).

3.2 CRBM and diffusion network

A diffusion network (DN) consists of n fully-connected units with activation $\{x_j(t)\}$, connected by $n \times n$ real-valued weights $\boldsymbol{W}$. The diffusion network (DN) [6, 11] exhibits continuous probabilistic behaviour, driven by a stochastic differential 'diffusion equation' for the time-evolution of the state x_j

$$dx_j(t) = \kappa_j\left(\sum_i w_{ij}\varphi_i(x_i(t)) - \rho_j x_j(t)\right) \cdot dt + \sigma \cdot dB_j(t) \tag{6}$$

where $1/\kappa_j$ and $1/\rho_j$ represent, in model terms, the input capacitance and resistance of neuron j. In terms of a discretised state update process, κ_j controls the changing rate of state $x_j(t)$ and ρ_j is a state-decay term. $dB_j(t)$ introduces a Brownian motion term [11] that provides an additional random element in the network's behaviour. The increment, $B_j(t + dt) - B_j(t)$, for such Brownian motion is thus a Gaussian random variable with zero mean and variance dt [11]. The discrete-time diffusion process for a finite time increment Δt is

$$\begin{aligned} x_j(t + \Delta t) = x_j(t) + \kappa_j \sum_i w_{ij}\varphi_i(x_i(t))\Delta t \\ - \kappa_j\rho_j x_j(t)\Delta t + \sigma z_j(t)\sqrt{\Delta t} \end{aligned} \tag{7}$$

where $z_j(t)$ is a Gaussian random variable with zero mean and unit variance. If $\Delta t = 1/\kappa_j\rho_j$, the terms in $x_j(t)$ cancel and writing $\sigma\sqrt{(\Delta t)} = \sigma'$, this becomes

$$x_j(t + \Delta t) = \kappa_j \sum_i w_{ij}\varphi_i(x_i(t))\Delta t + \sigma' z_j(t) \tag{8}$$

If $\boldsymbol{W}$ is symmetric and κ_j is a constant over the network, the right hand side of (8) is equivalent to the total input of a CRBM as given by (3). As $s_j = \varphi_j(x_j)$, the CRBM is simply a symmetrical restricted diffusion network. Thus, although the CRBM is a highly constrained diffusion network, the weight-update algorithm of the DN is suitable for training a CRBM.

3.3 MCD training algorithms for the CRBM

The training rule for any parameter λ_j in a DN is [6]

$$\Delta\lambda_j = \langle S_{\lambda_j} \rangle_0 - \langle S_{\lambda_j} \rangle_\infty \tag{9}$$

where $\langle \cdot \rangle_0$ and $\langle \cdot \rangle_\infty$ refer to the expectation values over the training data with visible states clamped, and in free-running equilibrium, respectively. S_{λ_j} is the system-covariate [6], which is the negative derivative of the DN's energy function with respect to parameter λ_j. Drawing on the relationship between the restricted DN and the product of experts [12], we simplify (9) by once again minimising contrastive divergence [2], avoiding the time-consuming process of full Gibbs' sampling to reach equilibrium. If $\langle \cdot \rangle_1$ is the expectation value over one-step sampled data, the MCD rule becomes

$$\Delta\hat{\lambda}_j = \langle S_{\lambda_j} \rangle_0 - \langle S_{\lambda_j} \rangle_1 \tag{10}$$

for both the restricted diffusion network and CRBM. The energy function of CRBM is analogous to that of the continuous Hopfield model [13]

$$U = -\frac{1}{2}\sum_{i \neq j} w_{ij} s_i s_j + \sum_j \frac{1}{a_j} \int_0^{s_j} \varphi^{-1}(s) ds \tag{11}$$

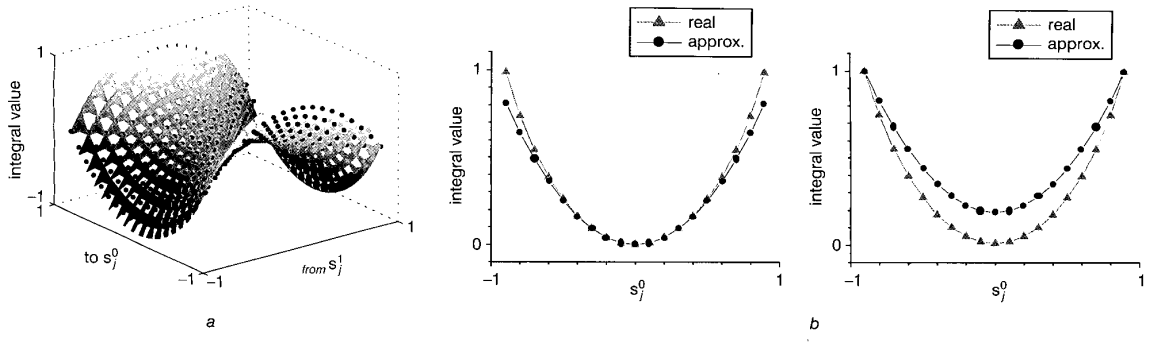

Fig. 2 *Approximation of (14)*

a Approximation of (14) over the value range of s_j^0 (=s_j) and s_j^1 (=$\hat{s}_j$)
b Two slices at $s_j^1 = 0$ (left) and $s_j^1 = -0.9$ (right). Note that, owing to the difference between the real and approximate values, the black points are invisible over some ranges

where $\varphi(s)$ refers to $\varphi_j(s)$ with $a_j = 1$. Equations (10) and (11) then lead to MCD training rules for both the CRBM's weights $\{w_{ij}\}$ and 'noise-control' parameters $\{a_j\}$.

$$\Delta\hat{w}_{ij} = \eta_w(\langle s_i s_j\rangle - \langle\hat{s}_i\hat{s}_j\rangle) \tag{12}$$

$$\Delta\hat{a}_j = \eta_a\left(\frac{1}{a_j^2}\left\langle\int_{\hat{s}_j}^{s_j}\varphi^{-1}(s)ds\right\rangle\right) \tag{13}$$

where $\hat{s}_j$ denotes, as before, the one-step sampled state of unit j, and $\langle\cdot\rangle$ in (13) refers to the mean over the training data. To simplify this, primarily for ease of hardware design, we approximate the integral term in (13) as

$$\int_{\hat{s}_j}^{s_j}\varphi^{-1}(s)ds \propto (s_j + \hat{s}_j)(s_j - \hat{s}_j) \tag{14}$$

The accuracy of this approximation is illustrated in Fig. 2. The horizontal axes in Fig. 2*a* span the full ranges of s_j and $\hat{s}_j$, the continuous surface shows the values of the real integral and the black points illustrate the approximate values given by (14). Fig. 2*b* shows two slices of Fig. 2*a* at $\hat{s}_j = 0$ and $\hat{s}_j = -0.9$, respectively. The difference is small and encourages the use of the approximation in (14). From (14) and (13), the simplified training rule for a_j is

$$\Delta\hat{a}_j = \frac{\eta_a}{a_j^2}(\langle s_j^2\rangle - \langle\hat{s}_j^2\rangle) \tag{15}$$

Equations (12) and (15) indicate that the training rules for CRBM require simple addition and multiplication operations. The similarity between these two equations is striking. Since the MCD training rule for $\{w_{ij}\}$ has been successfully implemented and tested in VLSI [14, 15], the similarity implies that these training rules for $\{w_{ij}\}$ and $\{a_j\}$ may be implemented with the same circuit. In addition, Alspector *et al.* have implemented a full Boltzmann machine in VLSI [16], providing potential solutions to all other component circuits a CRBM in VLSI will require.

4 Demonstration: artificial data

Simple, but nontrivial, two-dimensional datasets, shown in Fig. 3*a*, were generated to compare and illustrate the performance of RBM, RBMrate and CRBM on continuous (analogue) data. A typical dataset comprises two clusters of 200 data each. An RBM with six hidden units, an RBMrate model with six hidden units and a CRBM with four hidden units were trained for 4000 epochs with $\eta = \eta_w = 1.5$ in (2) and (12). The numbers of hidden units in each case were chosen empirically to maximise modelling ability while minimising the number of free parameters. To compare the trained generative models, an approximate equilibrium reconstruction by each of the models was obtained by Gibbs sampling from 400 random initial data for 20 steps. RBM's limited ability to model asymmetric data is clear from Fig. 3*b*. Fig. 3*c* shows clearly that the discrete stochastic units in RBMrate improve the model significantly, although the discrete-valued nature of the reconstruction is revealed by the 'graininess' in the generated data. The CRBM was trained with $\eta_a = 1$, $\theta_H = 1$, $\theta_L = -1$ and $\sigma = 0.2$ for all units. The reconstruction by the trained CRBM is shown in Fig. 3*d*. The CRBM models the data well and reconstructs continuous-valued points with no artificial quantisation.

Equations (15) and (12) were used to adapt the CRBM's 'noise-control' parameter $\{a_j\}$ and the weights $\{w_{ij}\}$ in parallel. Figs. 4*a* and *b* show the evolution of the $\{a_j\}$ for the visible and hidden units during training. These traces

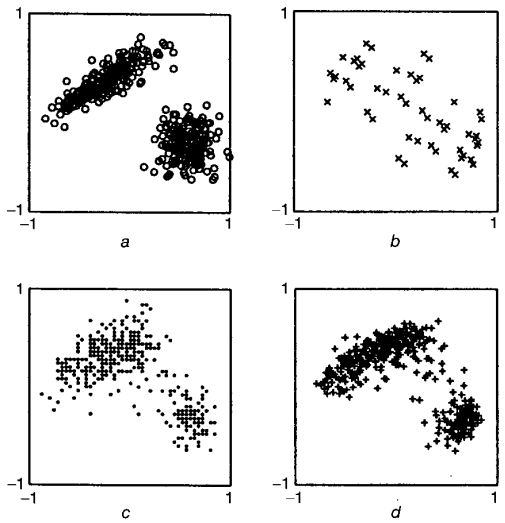

Fig. 3 *Artificially generated continuous training data and reconstruction from 400 random input by trained models after 20 steps of Gibbs sampling*

a Artificially generated continuous training data
b Reconstruction using RBM
c Reconstruction using RBMrate
d Reconstruction using CRBM

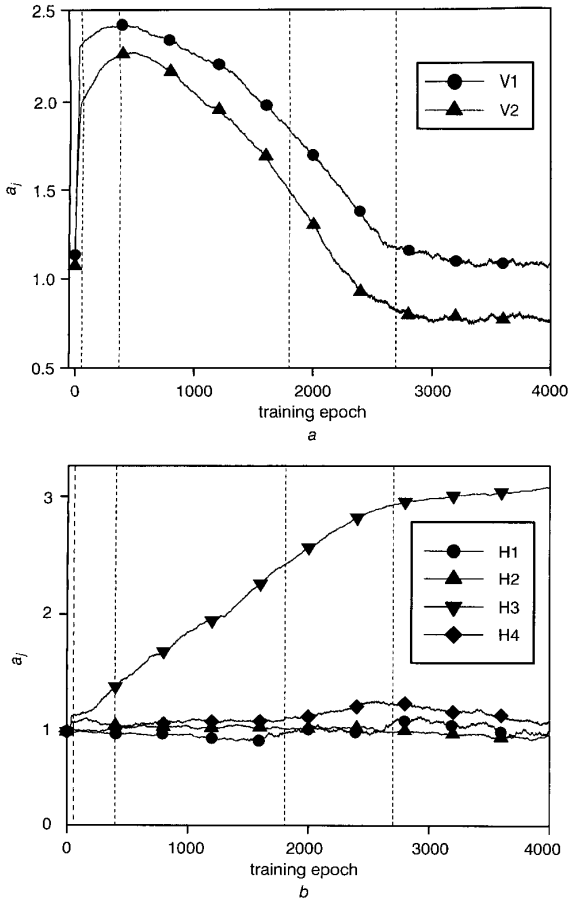

Fig. 4 *Evolution of 'noise-control' parameter a_j for visible and hidden units during CRBM training*

a Visible units
b Hidden units

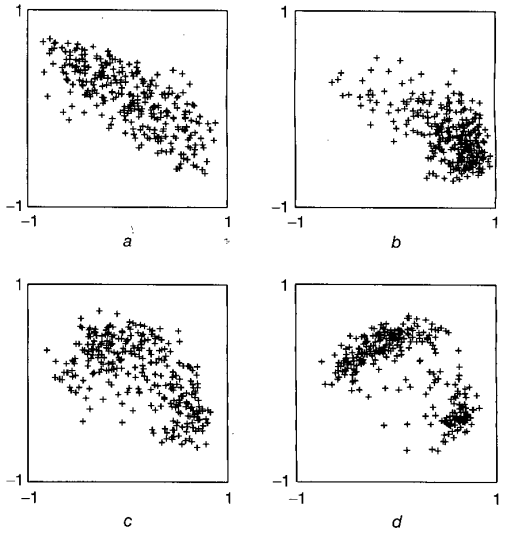

Fig. 5 *20-step Gibbs sampled reconstructions by the CRBM*

a After 50 training epochs
b After 395 training epochs
c After 1800 training epochs
d After 2700 training epochs

show clearly how the data model of Fig. 3*d* forms during training. a_j for hidden unit 1 rises from $a_1 = 1$ to $a_1 = 3$, while a_2, a_3 and a_4 remain ≈ 1, so hidden unit 1 and hidden units 2–4 are clearly performing different functions in the model. The $\{a_j\}$ of the visible units display a form of 'autonomous annealing' driven by (15), gradually reducing the level of noise and thus the units' tendency to be binary.

To investigate what this behaviour means in terms of model formation during training, 20-step reconstructions by the CRBM after 50, 395, 1800 and 2700 training epochs are shown in Fig. 5. The broken vertical lines in Fig. 4 highlight the corresponding values of $\{a_j\}$. Very early in training, the $\{a_j\}$ for all of the visible layer rises abruptly to form near-binary units, allowing a wide 'search' of the solution space to find crude binary-model approximations to the input data clusters, as illustrated by Fig. 5*a*. This model is similar to the binary RBM model in Fig. 3*b*. The $\{a_j\}$ of visible units peak at epoch 395. Fig. 5*b* reveals that the large value of $\{a_j\}$ compresses the reconstruction and inhibits the adaptation of the weights $\{w_{ij}\}$. The $\{a_j\}$ of visible units subsequently decreases, allowing more flexible adaptation of the $\{w_{ij}\}$. As training progresses further, the $\{a_j\}$ of visible units continues to fall, enabling the CRBM to model continuous data with more deterministic visible units. During this adaptation process, a_2 in the hidden layer rises, causing its unit to become a near-binary 'decision-maker' and allowing the two distinct clusters to be modelled clearly, as illustrated by the reconstruction in Figs. 5*c* and 5*d*. The other hidden units encode the variances of the clusters in different directions with smaller values of $\{a_j\}$. By epoch 2700, the values of $\{a_j\}$ in both hidden and visible units have essentially settled and the CRBM models the data distribution well, as shown in Fig. 5*d*. The weight parameters $\{w_{ij}\}$, though not shown, also settle after epoch 2700, indicating that the training reaches equilibrium. The CRBM finally reconstructs data successfully at epoch 4000, as shown in Fig. 3*d*. The results above indicate that the training equation (15) leads to intriguing and encouraging training behaviour in this stylised, but nontrivial example.

5 Demonstration: real heartbeat (ECG) data

To highlight the improved modelling richness of the CRBM and to give these results credence, the CRBM was trained to model the ECG data used in [4] and [17]. Training and test datasets, of 500 and 1700 heartbeats, respectively, were extracted from a 30-minute ECG trace. Each heartbeat consists of 65 sampled values with amplitudes normalised into the range $[-1, 1]$. The 500 training data contain six ventricular ectopic beats (VEBs) and the 1700 test data contain 27 VEBs.

A CRBM with four hidden units was trained for 4000 epochs with $\eta_w = 1.5$, $\eta_a = 1$, $\theta_H = 1$, $\theta_L = -1$, $\sigma = 0.2$ for visible units and $\sigma = 0.5$ for hidden units. Fig. 6 shows the 20-step reconstruction by the trained CRBM with input of a normal QRS complex (Fig. 6*a*), typical of $\approx$99% of the training data (The letters QRS refer to the different points in a heartbeat cycle when the heart is actually pumping current [17]) a typical VEB (Fig. 6*b*). The CRBM modelled both forms of heartbeat successfully, although VEBs represent only 1% of the training data. Following [4], Fig. 6*c* shows the receptive fields of the hidden bias unit and the four hidden units. The bias unit codes an 'average' normal QRS complex, and H3 adds amplitude and detailed structure to the P- and T-waves. H1 and H2

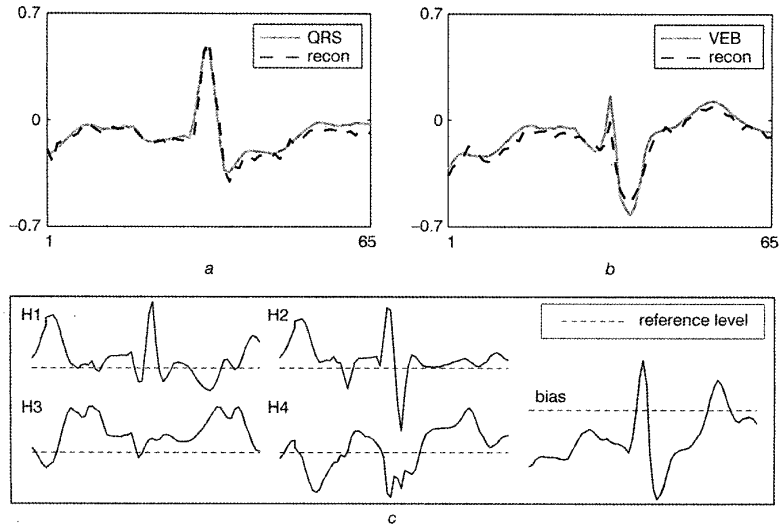

Fig. 6 *Reconstruction by the trained CRBM with four hidden units and the receptive fields of the hidden bias and the four hidden units*
a Reconstruction from input of normal QRS complex
b Reconstruction from input of typical VEB
c Receptive fields of the hidden bias and the four hidden units H1–H4

drive a small horizontal shift and a magnitude variation of the QRS complex. Finally, H4 encodes the significant dip found in a VEB. A CRBM with only two hidden units yielded similar success, although the ability to model inter-QRS variations was reduced. This modelling richness compares favourably with that of an RBM with six hidden units [4], which merely modelled a normal QRS complex.

The most principled measure of a test datum's fit to the CRBM model is the log-likelihood of the particular datum under the trained CRBM. Low likelihood indicates a VEB and extremely low likelihood suggests an artefact. However, the likelihood calculation requires complicated calculations, particularly in hardware. The receptive field of H4 suggests a simpler, pragmatic approach. When H4 is active, the dip characteristic of a VEB is included in the model and H4's activity should thus be an indicator of a VEB. In addition, parameter a_j for H4 ends with a large value after training, similar to that for H3 in Fig. 4*b*, transforming H4 into a binary 'decision maker'. Therefore, the activities of hidden units, in particular that of H4, may form the basis of a simple novelty detector. The post-sigmoid activities of H4 corresponding to 1700 test data are shown in Fig. 7, with the noise source in (3) removed ($\sigma = 0$). The strong peaks highlight the VEBs clearly. VL in Fig. 7 indicates the minimum H4 activity for a VEB in the test data, and QH marks the maximum H4 activity induced by a normal QRS heartbeat. A simple linear classifier with a threshold between the two dashed lines will therefore detect the VEBs with an accuracy of 100%.

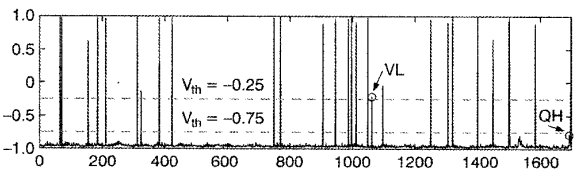

Fig. 7 *Post-sigmoid activities of H4 corresponding to 1700 test data*

The margin for this threshold is more than 0.5, equivalent to 25% of the total value range. Compared to [17] where a 64–16–64 autoassociative MLP was trained to model the same data, the best accuracy achieved was merely 99.3%. Wang *et al.* also indicates that most neural-network models can only distinguish VEB from normal heartbeats with an accuracy of no more than 97% [18], though various models were tested with various datasets. A single hidden unit's activity in a CRBM is, therefore, potentially a reliable novelty detector and it is expected that layering a supervised classifier on the CRBM, to 'fuse' the hidden unit activities, will lead to improved results.

Constant σ in (3) controls the noise variance of the continuous stochastic units. If the value of σ is very large such that the noise component dominates, the unit become 'binary-random' with a probability of 50%, independent of the deterministic input. Note that such a binary-random unit is different from the binary-stochastic unit induced by a large value of a_j. The probability of the binary-stochastic unit remains dependent on the deterministic input. However, the units become fully deterministic if $\sigma = 0$. To investigate the influence of the noise variance σ^2 on the CRBM's performance, CRBMs with various values of σ for the hidden units were trained to model the same ECG data, and the margin allowed to detect the VEBs with 100% accuracy used as the indicator of the trained CRBMs' performance. Fig. 8 shows the results for the CRBMs with two, four and eight hidden units. The maximum margin always occurs with an intermediate value of σ. The degradation of the CRBM's performance with smaller σ can be attributed to 'overfitting' by near-deterministic hidden units, while the degradation with larger σ is more simply attributable to the domination of input noise. Fig. 8 also reveals that increasing the number of hidden units does not necessarily improve the margin. A CRBM with eight hidden units 'uses' more than one hidden units to model the characteristics of VEBs, so the performance of a single hidden unit as a VEB-detector is reduced. Clearly, a parsimonious architecture and an optimum level of noise are key to the CRBM's performance as a novelty, or rare-event detector.

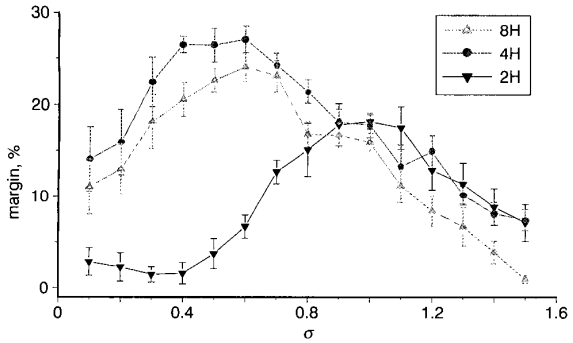

Fig. 8 *Influence of σ on CRBM performance*

Points and error bars show the means and standard deviations across ten experiments

6 Conclusions

The CRBM is a principled approach to modelling continuous data successfully with a simplified MCD rule. The architecture has been shown to be able to develop stochastic behaviour in its units, in response to training, that ranges from binary-stochastic to deterministic. This leads to a modelling flexibility that exceeds the ability of probabilistic models with more fixed-form (e.g. binary) units. This is all clear from experiments with artificial data designed to probe how a CRBM forms a generative model. Experiments with real ECG data confirm that the CRBM is capable of modelling complex continuous data, and we have also shown that the CRBM's hidden units can underpin a simple but reliable novelty detector. Component circuits of the RBM have been successfully implemented [9, 14, 15] and circuits for noise sources have also been demonstrated [16]. Therefore, the CRBM is a potential continuous stochastic model for VLSI implementation and embedded intelligent systems.

7 Acknowledgments

The authors are grateful to C. Williams for direct assistance and advice, and to Y.-W. Teh, T. Marks and J. Movellan for extensive email assistance.

8 References

1 HINTON, G.E.: 'Products of experts'. Proceedings of 9th Int. Conf. on Artificial neural networks, (ICANN'99), Edinburgh, Scotland, UK, Sept. 1999, pp. 1–6
2 HINTON, G.E.: 'Training products of experts by minimizing contrastive divergence'. Technical Report 2000, Gatsby Comp. Neuroscience Unit
3 SMOLENSKY, P.: 'Information processing in dynamical systems: foundations of harmony theory' in 'Parallel distributed processing: explorations in the microstructure of cognition' (MIT Press, Cambridge, MA, USA, 1986), Vol. 1, pp. 195–281
4 MURRAY, A.F.: 'Novelty detection using products of simple experts—a potential architecture for embedded systems', *Neural Netw.*, 2001, **14**, pp. 1257–1264
5 TEH, Y.W., and HINTON, G.E.: 'Rate-coded restricted Boltzmann machine for face recognition' in 'Advances in neural information processing system' (MIT Press, Cambridge, MA, USA, 2001), Vol. 13
6 MOVELLAN, J.R.: 'A learning theorem for networks at detailed stochastic equilibrium', *Neural Comput.*, 1998, **10**, pp. 1157–1178
7 CHEN, H., and MURRAY, A.F.: 'A continuous restricted Boltzmann machine with hardware-amenable learning algorithm'. Proceedings of 12th Int. Conf. on Artificial neural networks (ICANN2002), Madrid, Spain, Aug. 2002, pp. 358–363
8 HINTON, G.E., and SEJNOWSKI, T.J.: 'Learning and re-learning in Boltzmann machines' in 'Parallel distributed processing: explorations in the microstructure of cognition' (MIT Press, Cambridge, MA, USA, 1986), Vol. 1, pp. 283–317
9 WOODBURN, R.J., ASTARAS, A.A., DALZELL, R.W., MURRAY, A.F., and McNEILL, D.K.: 'Computing with uncertainty in probabilistic neural networks on silicon'. Proceedings 2nd Int. ICSC Symp. on Neural computation, 1999, pp. 470–476
10 FREY, B.J.: 'Continuous sigmoidal belief networks trained using slice sampling' in Advances in neural information processing systems' (MIT Press, Cambridge, MA, USA, 1997), Vol. 9, pp. 452–458
11 MOVELLAN, J.R., MINEIRO, P., and WILLIAMS, R.J.: 'A Monte-Carlo EM approach for partially observable diffusion processes: theory and applications to neural networks', *Neural Comput.*, 2002, **14**, pp. 1501–1544
12 MARKS, T.K., and MOVELLAN, J.R.: 'Diffusion networks, products of experts, and factor analysis'. Technical Report 2001.02., UCSD MPLab, 2001
13 HOPFIELD, J.J.: 'Neurons with graded response have collective computational properties like those of two-state neurons', *Proc. Natl. Acad. Sci. USA*, 1984, **81**, pp. 3088–3092
14 FLEURY, P., WOODBURN, R.J., and MURRAY, A.F.: 'Matching analogue hardware with applications using the products of experts algorithm'. Proceedings of IEEE European Symp. on Artificial neural networks, 2001, pp. 63–64
15 FLEURY, P., MURRAY, A.F., and REEKIE, M.: 'High-accuracy mixed-signal VLSI for weight modification in contrastive divergence learning'. Proceedings of 12th Int. Conf. on Artificial neural networks (ICANN2002), Madrid, Spain, August 2002, pp. 426–431
16 ALSPECTOR, J., GANNETT, J.W., HABER, S., PARKER, M.B., and CHU, R.: 'A VLSI-efficient technique for generating multiple uncorrelated noise sources and its application to stochastic neural networks', *IEEE Trans. Circuits Syst.*, 1991, **38**, pp. 109–123
17 TARASSENKO, L., and CLIFFORD, G.: 'Detection of ectopic beats in the electrocardiogram using an autoassociative neural network', *Neural Process. Lett.*, 2001, **14**, pp. 15–25
18 WANG, Y., ZHU, Y.S., THAKOR, N.V., and XU, Y.H.: 'A short-time multifractal approach for arrhythmia detection based on fuzzy neural network', *IEEE Trans. Biomed. Eng.*, Sept. 2001, **48**, pp. 989–995